

# FTP & TFTP

Leonardo H. Añez Vladimirovna\*

*Universidad Autónoma Gabriel René Moreno,  
Facultad de Ingeniería en Ciencias de la Computación y Telecomunicaciones,  
Santa Cruz de la Sierra, Bolivia*

2 de diciembre de 2019

## File Transfer Protocol (FTP)

Es uno de los protocolos estandar del Internet, utilizados para transferir archivos de datos entre un cliente y un servidor. Fue desarrollado a principios de los 70s por Abhay Bhushan (MIT). Se creó inicialmente para permitir la transferencia segura entre servidores y computadores del ARPANET.

### Funcionamiento

Tiene una arquitectura cliente/servidor proporcionando servicio a travez de dos puertos:

- **Port: 20** Para la transferencia de datos
- **Port: 21** Para control (órdenes)

El cliente se conecta al servidor desde un puerto superior al 1024 y hace una solicitud al servidor por el **Port: 21**, que siempre está escuchando als peticiones de los clientes por ese puerto. Una vez establecido se pueden usar las ordenes especificas.

### Usuarios

- **Autenticados:** Ingresan con nombre y contraseña. Dentro de estos, hay usuarios FTP y virtuales. Los primeros son usuarios del sistema y pueden acceder a partes del sistema con permisos. Los virtuales son los que tienen cuentas en BD como MySQL y solo se autentifican para utilizar FTP.
- **Anónimos:** No disponen de cuenta y para conectarse al servidor FTP introducen cuentas simbólicas. Es la forma convencional de llamar al servicio de transferencia de ficheros que las organizaciones dejan para acceso público. La clave mas habitual es **guest**.

## Modos de Conexión a un Servidor FTP

### Modo Activo

1. El cliente se conecta desde un puerto de control aleatorio superior al 1024 (en nuestro caso el 1035) al puerto de control del servidor, puerto 21. El cliente empieza a escuchar por el puerto 1036 y envía este puerto de control al servidor.
2. El servidor responde con un ACK al puerto de control del cliente.
3. El servidor inicia una conexión entre su puerto de datos (puerto 20) y el puerto de datos del cliente (puerto 1036)
4. El cliente responde con un ACK al servidor.

### Modo Pasivo

1. El cliente se conecta desde un puerto de control aleatorio superior al 1024 (en nuestro caso el 1035) al puerto de control del servidor, puerto 21. El cliente envía un comando PASV al servidor.
2. El servidor responde con un ACK, desde un puerto aleatorio superior al 1024 (en nuestro caso el 2040) al puerto de control del cliente.
3. El cliente inicia una conexión entre su puerto de datos del cliente (puerto 1036) y el puerto de datos del servidor (puerto 2040)
4. El servidor responde con un ACK al cliente.
5. Cerrar sesión.

---

\*Correo Electrónico: toborochi98@outlook.com

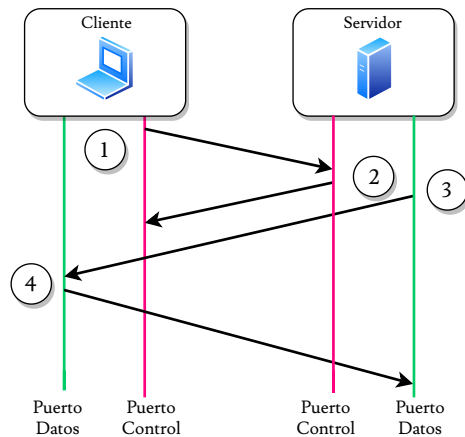


Figura 1: Activo

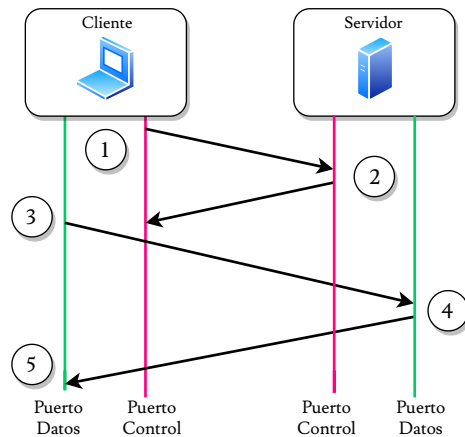


Figura 2: Pasivo

## Modos de Transferencia

- Modo ASCII: Se usa para transmitir texto y HTML.
- Modo Binario: Transfiere zips, imagenes, ejecutables. No se pueden enviar en modo ASCII ya que causarian problemas y viceversa.

## Servidores FTP

### Trivial File Transfer Protocol (TFTP)

Es un protocolo de transferencia simple semejante a una version basica de FTP y se utiliza para transferir archivos pequeños.

### Características

- Utiliza UDP en el puerto 69.
- No existen mecanismos de autenticación o cifrado.
- Se utiliza para leer o escribir ficheros en un servidor remoto.

Dado que UDP no es confiable, utiliza el tiempo de espera y retransmisión para garantizar que lleguen los datos. El lado emisor transmite un archivo de tamaño fijo (512), espera el ACK del bloque y aguarda un ACK por cada bloque antes de enviar el siguiente.

### Funcionamiento

Una máquina A, que inicia la comunicación, envía un paquete RRQ (Read Request) o WRQ (Write Request) a una maquina B, con el nombre del archivo y el modo de transferencia. B responde con un ACK, que tambien sirve para informar a A del puerto que en B al que tiene que enviar los paquetes restantes. La maquina origen envia paquetes de datos numerados al destino, excepto el último conteniendo 512Bytes. La del destino le responde con ACK. El paquete final debe contener menos de 512 de datos para indicar que es el último, si el ultimo es de 512Bytes exactos, se envia un siguiente con 0Bytes.